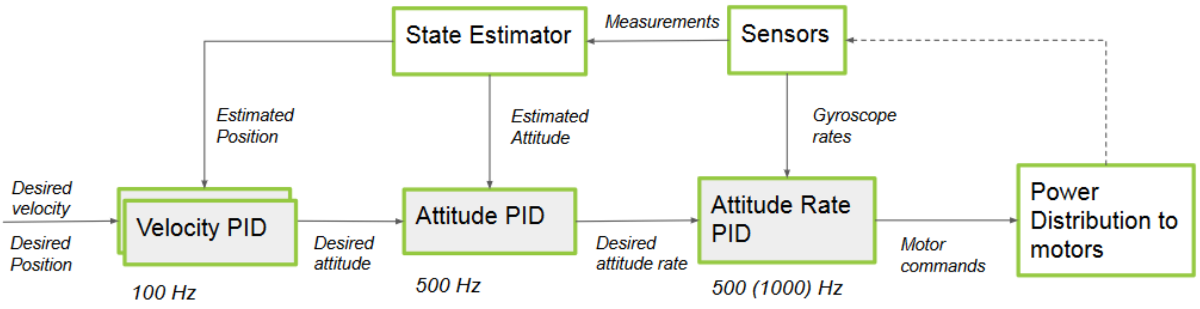
PID tuning the crazyflie

The present document explains the steps required to tune the crazyflie's PID. As the crazyflie firmware is in constant evolution this may become obsolete. Written date : 06.08.20

There are three cascaded PIDs : rate, attitude, velocity/position.



We wills focus on the rate and attitude ones for roll and pitch as they are more important to get a stabilized flying drone.

Preliminary steps

We want to first deactivate the crash detection (situation awareness). The drone will often turn upside down while tuning the rate PID, and having to reset it each time would be annoying.

In crazyflie-firmware/src/modules/interface/sitaw.h : the lines #define SITAW_ENABLED and #define SITAW_TU_ENABLED should be commented.

In crazyflie-firmware/src/modules/interface/sitaw.c :

```
static trigger_t sitAwFFAccWZ;

static trigger_t sitAwARAccZ;

static trigger_t sitAwTuAcc;
```

should be commented to avoid compilation errors.

Then we set to zero all PID gains for rate and attitude no need to change the integration limit. The latter are defined in crazyflie-firmware/src/modules/interface/pid.h.

As we first tune the rate PID, we need to switch the control mode. In crazyflie-firmware/src/modules/src/crtp_commander_rpyt.c :

```
static RPYType stabilizationModeRoll = ANGLE;
static RPYType stabilizationModePitch = ANGLE;
```

should become

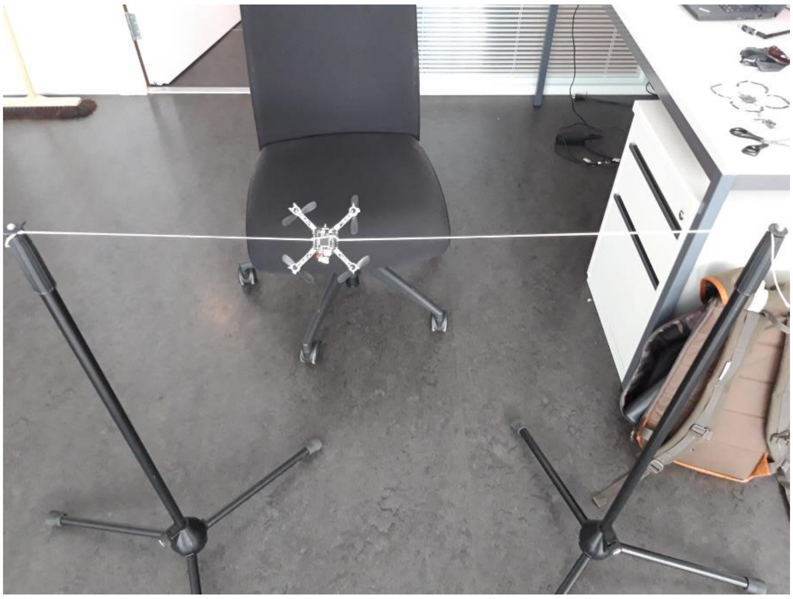
```
static RPYType stabilizationModeRoll = RATE;
static RPYType stabilizationModePitch = RATE;
```

Once all of this is done the modified firmware is uploaded to the crazyflie.

Then we connect to the crazyflie using the client. In the “parameter” tab we can access and change directly the PID gains. Parameter names are pid_rate and pid_attitude. A gaming controller is used to control the crazyflie.

Tuning the rate PID

To tune the rate PID the crazyflie should be attached with a string.



With this setup the drone can freely vary its pitch, therefore we will tune it and then copy paste the gains for the roll.

First we increase the proportional gain by steps and try to rotate the drone. Once it becomes unstable (oscillating behavior) we go back to the previous stable gain.

We repeat the same thing for KI.

KD stays 0 except if it feels like the drone is overshooting too much.

Tuning the attitude PID

To tune the attitude PID we restore crazyflie-firmware/src/modules/src/crtp_commander_rpyt.c so that we control the attitude again. The situation awareness should be reenabled as the drone will have to fly freely.

Same steps as for the rate tuning except that the drone is no longer hanging.

Final step

Now that all the gains are known they should be written in PID.h.